

Google Cloud REST API

API Documentation

URL LINK	2
DATA MODEL	3
GET ALL USERS	4
CREATE A BOAT	5
GET A BOAT	6
LIST ALL BOATS	8
EDIT A BOAT WITH PUT	10
EDIT A BOAT WITH PATCH	12
DELETE A BOAT	14
CREATE A LOAD	15
GET A LOAD	17
LIST ALL LOADS	18
EDIT A LOAD WITH PUT	20
EDIT A LOAD WITH PATCH	22
DELETE A LOAD	24
ASSIGN LOAD TO BOAT	25
REMOVE LOAD FROM BOAT	26
METHODS NOT ALLOWED	27

URL Link

- URL link for application deployed on GCP and account creation/login.

* Note: The following link is not functional, only for demonstration purposes.

<https://project-klemzcea.ue.r.appspot.com>

Data Model

The app stores three kinds of entities in Datastore, Boats, Loads and Users.

Boats

Property	Data Type	Notes
id	Integer	The id of the boat. Datastore automatically generates it.
name	String	Name of the boat.
type	String	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	The length of the boat in feet.
loads	List	Holds the ids of the loads the boat is carrying. A boat can have many pieces of load.
owner	String	The id of the user/owner. The id is the sub attribute of JWT. A boat can only have one owner, but an owner can have multiple boats.

Loads

Property	Data Type	Notes
id	Integer	The id of the load. Datastore automatically generates it.
volume	Integer	The volume of the load.
carrier	Dictionary	Contains the id of the boat that is carrying the load. A load can only be on one boat.
item	String	The name of the item in the load.
creation_date	String	Date the load was created

Users

Property	Data Type	Notes
user_id	String	The id of the user. The id is the sub attribute of JWT. A boat can only have one owner/user, but an owner can have multiple boats.

Get all Users

View all users

GET /users

Request

Path Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: JSON only

Failure: JSON only

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	No Accept: application/json is header included

Response Examples

Success

Status: 200 OK

```
{
  {
    "user_id": "112864445118289180073"
  },
  {
    "user_id": "112483434652610161194"
  }
}
```

Failure

Status: 406 Not Acceptable

```
{
  "Error": "Accept header must be application/json"
}
```

Create a Boat

Allows you to create a new boat. A newly created boat will not have any loads.

When a boat is created, the owner of the boat will contain the user id (sub attribute of the current JWT). This will create a relationship between the User entity and the Boat (non-user) entity.

A boat can only have one owner, but an owner can have multiple boats.

A JWT token needs to be provided in the Authorization. Protected endpoint.

POST /boats

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	Yes
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Length of the boat in feet.	Yes

Request Body Example

```
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28
}
```

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	The boat created will not be carrying any load.
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code will be returned.
Failure	401 Unauthorized	If no JWT is provided(missing) or JWT is invalid
Failure	406 Not Acceptable	No Accept: application/json header is included

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The value of the attribute `owner` contains the user id (sub attribute of current JWT).

Success

Status: 201 Created

```
{
  "id": 6234586254147584,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "owner": "112483434652610161194",
  "self": "https://project-klemzcea.ue.r.appspot.com/boats/6234586254147584"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

Status: 401 Unauthorized

```
{
  "Error": "JWT token is either missing or invalid"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Accept header must be application/json"
}
```

Get a Boat

Allows you to get/view an existing boat owned by a user.

A JWT token needs to be provided in the Authorization. Protected endpoint.

GET /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	401 Unauthorized	If no JWT is provided(missing) or JWT is invalid
Failure	406 Not Acceptable	No Accept: application/json header is included
Failure	403 Forbidden	Cannot view this boat because it is owned by someone else.

Response Examples

Success

```
Status: 200 OK
{
  "owner": "112483434652610161194",
  "length": 28,
  "loads": [],
  "type": "Catamaran",
  "name": "Sea Witch",
  "id": 5970178269511680,
  "self": "https://project-klemzcea.ue.r.appspot.com/boats/5970178269511680"
}
```

Failure

```
Status: 404 Not Found

{
  "Error": "No boat with this boat_id exists"
}

Status: 401 Unauthorized
{
  "Error": "JWT token is either missing or invalid"
}

Status: 406 Not Acceptable
{
  "Error": "Accept header must be application/json"
}

Status: 403 Forbidden
```

```
{
  "Error": "Cannot view this boat. Boat is owned by someone else"
}
```

List all Boats

List/view all the boats owned by a user

A JWT token needs to be provided in the Authorization. Protected endpoint.

GET /boats

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	If there are more than 5 boats to be viewed, a pagination link "next" will appear to view the remaining boats.
Failure	401 Unauthorized	If no JWT is provided(missing) or JWT is invalid
Failure	406 Not Acceptable	No Accept: application/json header is included

Response Examples

- Pagination was implemented using offsets and limits. The limit is to display 5 boats per page.

Success

Status: 200 OK

```
{
  "boats": [
    {
      "owner": "112483434652610161194",
      "name": "Tug Tug",
      "loads": [],
      "type": "Tugboat",
      "length": 12,
      "id": 5105806370406400,
      "self": "https://project-klemzcea.ue.r.appspot.com/boats/5105806370406400"
    },
    {
      "name": "Odessy",
      "length": 18,
      "owner": "112483434652610161194",
      "loads": [],
      "type": "Sail Ship",
      "id": 5647975057457152,
      "self": "https://project-klemzcea.ue.r.appspot.com/boats/5647975057457152"
    },
    {
      "name": "Spiky Tuna",
      "loads": [
        {

```

```

        "id": 5761076981399552,
        "self": "https://project-klemzcea.ue.r.appspot.com
        /loads/5761076981399552"
      },
      {
        "type": "Fishing Boat",
        "length": 35,
        "owner": "112483434652610161194",
        "id": 5668756323827712,
        "self": "https://project-klemzcea.ue.r.appspot.com
        /boats/5668756323827712"
      },
      {
        "name": "Colossus",
        "owner": "112483434652610161194",
        "length": 105,
        "loads": [],
        "type": "Cargo Ship",
        "id": 5678927947235328,
        "self": "https://project-klemzcea.ue.r.appspot.com
        /boats/5678927947235328"
      },
      {
        "length": 42,
        "type": "Yatch",
        "loads": [],
        "name": "Diamond of the Sea",
        "owner": "112483434652610161194",
        "id": 5682874216873984,
        "self": " https://project-klemzcea.ue.r.appspot.com
        /boats/5682874216873984"
      }
    ],
    "total_number_of_boats": 6,
    "next": "https://project-klemzcea.ue.r.appspot.com/boats?limit=5&offset=5"
  }
}

```

Failure

Status: 401 Unauthorized

```

{
  "Error": "JWT token is either missing or invalid"
}

```

Status: 406 Not Acceptable

```

{
  "Error": "Accept header must be application/json"
}

```

Edit a Boat with PUT

Allows you to edit a Boat.

A JWT token needs to be provided in the Authorization. Protected endpoint.

PUT /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	Yes
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Length of the boat in feet.	Yes

Request Body Example

```
{
  "name": "Titanic",
  "type": "Cruise Ship",
  "length": 100
}
```

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Boat was successfully updated.
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code will be returned.
Failure	401 Unauthorized	If no JWT is provided(missing) or JWT is invalid
Failure	406 Not Acceptable	No Accept: application/json header is included
Failure	403 Forbidden	Cannot edit this boat because it is owned by someone else.
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 200 OK
{
  "id": 5720442564247552,
```

```
    "name": "Titanic",
    "type": "Cruise Ship",
    "length": 100,
    "loads": [],
    "owner": "112483434652610161194",
    "self": "https://project-klemzcea.ue.r.appspot.com/boats/5720442564247552"
  }
```

Failure

Status: 400 Bad Request { "Error": "The request object is missing at least one of the required attributes" }
Status: 401 Unauthorized { "Error": "JWT token is either missing or invalid" }
Status: 406 Not Acceptable { "Error": "Accept header must be application/json" }
Status: 403 Forbidden { "Error": "Boat is owned by someone else" }
Status: 404 Not Found { "Error": "No boat with this boat_id exists" }

Edit a Boat with PATCH

Allows you to edit a Boat.

A JWT token needs to be provided in the Authorization. Protected endpoint.

```
PATCH /boats/:boat_id
```

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	Yes or No
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes or No
length	Length of the boat in feet.	Yes or No

Comment

- The request body requires at least one attribute.

Request Body Example

```
{
  "name": "Barbarossa",
  "type": "Pirate Ship"
}
```

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Boat was successfully updated.
Failure	400 Bad Request	If the request does not have at least one attribute or if there are more than 3 attributes.
Failure	401 Unauthorized	If no JWT is provided(missing) or JWT is invalid
Failure	406 Not Acceptable	If no Accept: application/json header is included
Failure	403 Forbidden	Cannot edit this boat because it is owned by someone else.
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 200 OK
{
  "id": 5720442564247552,
  "name": "Barbarossa",
  "type": "Pirate Ship",
  "length": 100,
```

```
"loads": [],  
"owner": "112483434652610161194",  
"self": "https://project-klemzcea.ue.r.appspot.com/boats/5720442564247552"  
}
```

Failure

Status: 400 Bad Request { "Error": "The request object is missing at least one of the required attributes" }
Status: 401 Unauthorized { "Error": "JWT token is either missing or invalid" }
Status: 406 Not Acceptable { "Error": "Accept header must be application/json" }
Status: 403 Forbidden { "Error": "Boat is owned by someone else" }
Status: 404 Not Found { "Error": "No boat with this boat_id exists" }

Delete a Boat

Allows you to delete a boat. If the boat is carrying a load, deleting the boat will unload any loads that were loaded on to it.

A JWT token needs to be provided in the Authorization. Protected endpoint.

```
DELETE /boats/:boat_id
```

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No boat with this boat_id exists
Failure	401 Unauthorized	If no JWT is provided(missing) or JWT is invalid
Failure	403 Forbidden	Cannot delete this boat because it is owned by someone else.

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found { "Error": "No boat with this boat_id exists" }
Status: 401 Unauthorized { "Error": "JWT token is either missing or invalid" }
Status: 403 Forbidden { "Error": "Boat is owned by someone else" }

Create a Load

Allows you to create a new load. All newly created loads are not assigned to any boat.

A load can only be on one boat.

POST /loads

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes
item	The name of item in the load	Yes
creation_date	Date the load was created	Yes

Request Body Example

```
{
  "volume": 5,
  "item": "LEGO Blocks",
  "creation_date": "10/18/2021"
}
```

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing 3 attributes, the load must not be created, and 400 status code must be returned.
Failure	406 Not Acceptable	If no Accept: application/json header is included

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The value of the attribute `carrier` is the ID of the boat currently carrying this load. Newly created loads have `carrier` set to null. If the load is not being carried by any boat, the value of `carrier` should be null.

Success

Status: 201 Created

```
{
  "carrier": null,
  "creation_date": "10/18/2021",
  "id": 5472288480690176,
  "item": "LEGO Blocks",
  "self": "https://project-klemzcea.ue.r.appspot.com/loads/5472288480690176",
  "volume": 5
}
```

Failure

Status: 400 Bad Request

```
{  
  "Error": "The request object is missing at least one of the required attributes"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Accept header must be application/json"  
}
```

Get a Load

Allows you to get an existing load.

```
GET /loads/:load_id
```

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

None

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	If no Accept: application/json header included

Response Examples

Success

Status: 200 OK

```
{
  "carrier": null,
  "creation_date": "10/18/2021",
  "volume": 5,
  "item": "LEGO Blocks",
  "id": 5472288480690176,
  "self": "https://project-klemzcea.ue.r.appspot.com/loads/5472288480690176"
}
```

Failure

Status: 404 Not Found

```
{
  "Error": "No load with this load_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Accept header must be application/json"
}
```

List all Loads

List/view all the loads.

GET /loads

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	If there are more than 5 loads to be viewed, a pagination link "next" will appear to view the remaining loads.
Failure	406 Not Acceptable	If no Accept: application/json header is included

Response Examples

- Pagination was implemented using offsets and limits. The limit is to display 5 loads per page.

Success

Status: 200 OK

```
{
  "loads": [
    {
      "carrier": {
        "id": 5647975057457152,
        "self": "https://project-klemzcea.ue.r.appspot.com/boats/5647975057457152"
      },
      "volume": 89,
      "item": "Canned goods",
      "creation_date": "01/13/2022",
      "id": 4907435588321280,
      "self": "https://project-klemzcea.ue.r.appspot.com/loads/4907435588321280"
    },
    {
      "item": "Jasmine Rice",
      "carrier": {
        "id": 5647975057457152,
        "self": "https://project-klemzcea.ue.r.appspot.com/boats/5647975057457152"
      },
      "creation_date": "23/04/2022",
      "volume": 33,
      "id": 5687010538815488,
      "self": "https://project-klemzcea.ue.r.appspot.com/loads/5687010538815488"
    },
    {
      "creation_date": "06/01/2022",
      "carrier": null,
      "item": "Oranges",
      "volume": 1000,
      "id": 5744039651442688,
      "self": "https://project-klemzcea.ue.r.appspot.com/loads/5744039651442688"
    }
  ],
}
```

```

{
  "item": "LEGO Blocks",
  "creation_date": "10/18/2021",
  "volume": 5,
  "carrier": {
    "id": 5720442564247552,
    "self": "https://project-klemzcea.ue.r.appspot.com/boats/5720442564247552"
  },
  "id": 5953111277436928,
  "self": "https://project-klemzcea.ue.r.appspot.com/loads/5953111277436928"
},
{
  "item": "Wood Planks",
  "creation_date": "12/31/2021",
  "volume": 250,
  "carrier": null,
  "id": 6549916444786688,
  "self": "https://project-klemzcea.ue.r.appspot.com/loads/6549916444786688"
}
],
"total_number_of_loads": 6,
"next": "http://127.0.0.1:8080/loads?limit=5&offset=5"
}

```

Failure

Status: 406 Not Acceptable

```

{
  "Error": "Accept header must be application/json"
}

```

Edit a Load with PUT

Allows you to edit a Load.

PUT /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes
item	The name of item in the load	Yes
creation_date	Date the load was created	Yes

Request Body Example

```
{
  "volume": 25,
  "item": "Canned Peaches",
  "creation_date": "06/01/2022"
}
```

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Boat was successfully updated.
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the load is not created, and 400 status code will be returned.
Failure	406 Not Acceptable	No Accept: application/json header included
Failure	404 Not Found	No load with this load_id exists

Response Examples

Success

```
Status: 200 OK
{
  "creation_date": "06/01/2022",
  "volume": 25,
  "carrier": null,
  "item": "Canned Peaches",
  "id": 5953111277436928,
  "self": "https://project-klemzcea.ue.r.appspot.com/loads/5953111277436928"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing or has more attributes than required"
}

Status: 406 Not Acceptable
{
  "Error": "Accept header must be application/json"
}

Status: 404 Not Found
{
  "Error": "No load with this load_id exists"
}
```

Edit a Load with PATCH

Allows you to edit a Load.

PATCH /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes or No
item	The name of item in the load	Yes or No
creation_date	Date the load was created	Yes or No

Comment

- The request body requires at least one attribute.

Request Body Example

```
{
  "volume": 100,
  "item": "Coffee"
}
```

Response

Response Body Format

JSON only

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Boat was successfully updated.
Failure	400 Bad Request	If the request does not have at least one attribute or if there are more than 3 attributes.
Failure	406 Not Acceptable	No Accept: application/json header included
Failure	404 Not Found	No load with this load_id exists

Response Examples

Success

```
Status: 200 OK
{
  "id": 5953111277436928,
  "volume": 100,
  "item": "Coffee",
  "creation_date": "06/01/2022",
  "carrier": null,
  "self": "https://project-klemzcea.ue.r.appspot.com/loads/5953111277436928"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing or has more attributes than required"
}

Status: 406 Not Acceptable
{
  "Error": "Accept header must be application/json"
}

Status: 404 Not Found
{
  "Error": "No load with this load_id exists"
}
```

Delete a Load

Allows you to delete a load. Deleting a load updates the boat that was carrying it (removes the load).

DELETE /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found
<pre>{ "Error": "No load with this load_id exists" }</pre>

Assign Load to Boat

A load is place in a boat.

PUT /boats/:boat_id/loads/:load_id

Request

Path Parameters

Name	Description
boats	Collection of boats
boat_id	ID of the boat
loads	Collection of loads
load_id	ID of the load

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and the load is not already loaded on another boat.
Failure	403 Forbidden	If a load is already assigned to one boat and then is assigned to another boat without first being removed.
Failure	404 Not Found	The specified boat and/or load does not exist.

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden

```
{
  "Error": "The load is already loaded on another boat"
}
```

Status: 404 Not Found

```
{
  "Error": "The specified boat and/or load does not exist"
}
```

Comment

- A load can only be on one boat, but a boat can have many loads.

Remove Load from Boat

Remove a load from a boat.

DELETE /boats/:boat_id/loads/:load_id

Request

Path Parameters

Name	Description
boats	Collection of boats
boat_id	ID of the boat
loads	Collection of loads
load_id	ID of the load

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and if this load_id is loaded in the boat.
Failure	404 Not Found	No boat with this boat_id is loaded with the load with this load_id.
Failure	403 Forbidden	If a load is not assigned to this boat, it cannot be removed.

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found { "Error": "The specified boat and/or load does not exist" }
Status: 404 Not Found { "Error": "This load is not assigned to this boat, and therefore cannot be removed" }

Methods Not Allowed

The method Delete is not allowed on the following endpoints.

DELETE /boats
DELETE /loads
DELETE /user

Request

Path Parameters

None

Request Body

None

Response

No body

Response Body Format

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Failure	405 Method Not Allowed	This method is not allowed at this endpoint

Response Examples

Failure

Status: 405 Method Not Allowed

```
{  
  "Error": "Method not allowed at this endpoint"  
}
```